Some Results on Learning

B.K. Natarajan

CMU-RI-TR-89-6-

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

February 1989

Table of Contents

Abstract

This paper presents some formal results on learning. In particular, it concerns algorithms that learn sets and functions from examples. We seek conditions necessary and suffident for learning over a range of probabilistic models for such algorithms.

1. Introduction

This paper concerns algorithms that learn sets and functions from examples for them. The results presented in this paper appeared in preliminary form in [Nafarajan, 1986; 1988]. The motivation behind the study is a need to better understand the class of problems known as "concept learning problems" in the Artificial Intelligence literature.

What follows is a brief definition of concept (or set) learning. Let £ be the (04) alphabet, r the set of all strings on Z, and for any positive integer *n,* 2? the set of strings on L of length *n.* Let/denote a subset of 2T and *F* a set of such subsets. An example for/ is a pair (xj), *e £\ je L, such that xe / iff *y=l.* Informally, a learning algorithm for *F* is an algorithm that does the following: given a sufficiently large number of randomly chosen examples for any set/ e F, the algorithm identifies a set *g e* F_t such that *g* is a good approximation of/ (These notions will be formalized later.) The primary aim of this paper is to study the relationship between the properties of *F* and the number of examples necessary and sufficient for any learning algorithm tor it.

To place this paper in perspective: There are numerous papers on the concept learning problem in the artificial intelligence literature. See [Michalski et at., 1983] for an excellent review. Much of this work is not formal in approach. On the other hand, many formal studies of related problems were reported in the inductive inference literature. See [Angiuin & Smith, 1983] for an excellent review. As it happened, the wide gap between the basic assumptions of inductive inference on the one hand, and the needs of the empiricists on the other, did not permit the formal work significant practical import. More recently, [Valiant, 1984] introduced a new formal framework for the problem, with a view towards probabilistic analysis. The framework appears to be of both theoretical and practical interest, and the results of this paper are based on it and its variants. Related results appear in [Angiufn, 1987; Rivest & Schapire, 1987; Berman & Roos, 1987; Laird, 1986; Keams et a!., 1986] amongst others. [Blumer et at., 1986] present an independent development of some of the results presented in this paper, their proofs hinging on some classical results in probability theory, while ours are mostly combinatorial in flavour.

We begin by describing a formal model of learning, our variant of the model first presented by [Valiant, 1984]. Specifically, we define the notion of polynomial leamabtHty of sets in Section 2. We then discuss the notion of asymptotic dimension of a family of concepts, ami use it to obtain necessary and sufficient conditions for leamabiHty. In doing so, we give a general learning algorithm that turns out to be surprisingly simple, though provably good. Section 3 deals with a slightly different learning model, one in which the learner is required to learn with one-skied error, i.e., his approximation to the set to be teamed must be conservative in that it is a subset of the set to be teamed. Section 4 deals with the time complexity of learning, identifying necessary and sufficient conditions for efficient learning. Section 5 generalizes the teaming model to consider functions instead of sets, instead of sets. Notions of asymptotic ieamabtlity and asymptotic dimension are defined in this setting ami necessary ana sufficient conditions for teamabtlrty obtained. This requires us to prove a rather interesting combinatorial result called the generalized shattering lemma. Finally, Section 6 deaSs with a non-asymptotic model of teaming, where the division is between finite and infinite, rather than on asymptotic behaviour. In particular, we consider learning sets and functions on the reals, introducing the notion of finite-learnability. We review the elegant results of [Blumer et ai., 1986] on conditions necessary and sufficient for leamabirity in this setting. We then identify conditions necessary and sufficient for the finite-learnability of functions on the reals.

2. Feasible Learnability of Sets

We begin by describing our variant of the learning framework proposed by [Valiant, 1984].

Let I be the binary alphabet (0,1), Z* the set of all strings on X, and for any positive integer n, let Σ^{n} be the set of strings of length *n* or less in £*. A *concept** *f* is any subset of I*. Associated with each concept/is the *membership functionf*2.~** {0,1}, such that/*(x)» I iff *x* e /. Unless otherwise required, we will drop the superscript in/[91] and use/to refer both to the function and to the set. An *example* for a concept is a pair *(xy\ xelT.ye* {0,1} such that *y* =/(x). A *family* of concepts *F* is any set of concepts on E\ A *learning algorithm* (or more generally, a learning function) for the family F, is an algorithm that attempts to infer approximations to a concept in *F* from examples for it. The algorithm has at its disposal a subroutine EXAMPLE, which when called returns a randomly chosen example for the concept to be learned. The example is chosen randomly according to an arbitrary and unknown probability distribution *P* on 2T, in that the probability that a particular example *(x/(x))* will be produced at any call of EXAMPLE is

Defn: Let / be a concept and *n* any positive integer. The projection f_n of / on 27[1]" is given by f_n = *fnZT.*

Defn: Let 5 be any set **A** sequence on *S* is simply a sequence of elements *oi S.* S^l denotes the set of all sequences of length / on *S,* while X(S) denotes the set of ai! sequences of finite length on *S.*

Defn: Let/be a concept on 2T and *P* a probability distribution on IT. A *sample* of size / for/with respect to *P* is a sequence of the form (X^JCJ)), *{x^{x^))^.JixtJ{x^)* where x_v x^.., x_t is a sequence of elements of Z\ randomly and independently chosen according to P.

Defn: Let/and *g* be any two sets. The symmetric difference of/and g_t denoted by/Ag, is defined by $f\Delta g = (f-g)\cup(g-f)$.

With these supporting definitions in hand, we present our main definition, intuitively, we will call a family *F* feasibly learnable if it can be learned from polynomial^ few examples, polynomial in an error parameter *h* and a length parameter *n.* The length parameter *n* controls the length of the strings the concept is to be approximated on, and the error parameter *h* controls the error allowed in the learnt approximation.

Defn: Formally, a family *F* is *feasibly leamabie* if there exists an algorithm[2] A such that

(a) A takes as input two integers *n* and A, where *n* is the size parameter, and *h* is the error parameter.

(h)A makes polynomial^ few calls of EXAMPLE, polynomial in *n* and A. EXAMPLE returns examples for some/€ *F,* where the examples are chosen randomly ami independently according

'we us© to term corjcept instead of a set to oortionm wWi. t» aifcW bte^^nce literature.

[2]Unless stated otherwise, by "algorithm" we mean a *tMt&fy representsible* procedure, not necessIty computable. That *m,* fte procedure might use we^-defhed *but* non-computable *functions m* primitives.

to an arbitrary and unknown probability distribution *P* on *I!**,*

(c) For alt concepts/€ *F* and all probability distributions *P* on Z*-, with probability (1-I/A), *A* outputs a concept *ge F* such that

$$\sum_{x \in f \Delta g} P(x) \leq Vh$$

Oefn: Let N be the set of natural numbers. The *learning function* *P:NxNxX(£*x{O,I})-*F associated with a learning algorithm A is defined as follows.

Learning Function ¥
Input *, ^integers; C: sample;
begin
Let $C = (x$ ...
Run *A* on inputs *njk*
In place of EXAMPLE, at the ^ call of EXAMPLE by A,
give *A (xtf;)* as example.
Output A's output.
end

We now introduce a measure caied the dimension tor a family of concepts. Recall that we defined the projection/$_B$ of/on I^{11} by/$_w$ » *tfriF)* Similarly, the projection F_m of the family *F* on Z* is given by F_n = **$\{f_n | f \in F\}$. We call F_n the n^{th}-subfamily of *F*.**

Defn: The *(Mmemfcm (A* a subfamily f^ demoted Iv d & i ^) fe deined t^
***dim**iQFJalogjflFJ).*
(Notation: For a set *X,* ffl denotes the cardinality, while for a string x, bd denotes the string length.)

Defn: Let ii:N-^N be a function of one variable, where N is the natural numbers. The *asymptotic dimension* (or more simply the dimension) erf a family *F* is *d(n)* if <&»</;) = 8(^(n)). That is, there exists a constant c such that

V *n : dimiFJ £ d(n)*

and *dim(FJ* ≥ *cd(n)* infinitely often.

We denote the asymptotic dimension of a famfly *F* by *dim(F).* We say a family F is of polynomial dimension if me asymptotic dimension of *F* is a polynomial in *n.*

With these definitions In hand, we can give our first result The resutt is a lemma concerning the *notion* of shattering. LetFbeafamByof subsetscf set*. We say that *F shatters* a set *S^X, if* for every Sj c 5$_f$ there exists/e F ^1^1 that/nS = S_v To mir krowtedge, this notion was first introduced by [Vapni< & Chervonertcis, 19711.

We can now state our first result.

Lemma 1 (Shattering LMIHW;) v F_m *m* erf cftroreion * then F_H shatters a set of **size** [3]***ceiling(d(n+2))*** « Ate), w « y set shalened by F is of size at most *£

---

[3] ***ceiling(r)* is the least integer greater than *r*.**

Proof: First, we prove the upper bound. Suppose a set S is shattered by by F_n. Since there are 2^{151} distinct subsets of F_n, it follows from the definition of shattering that $2®$ <> $\backslash F_n \backslash$. Taking logarithms on both sides of the inequality, we get 151 £ $logQFJ) = d_{\%}$ which is as desired. To prove that the upper bound can attained, simply let F be all possible subsets of some d strings in r*~.

We prove the lower bound part of the lemma through the following claim. A variant of the claim is given by Vapnik & Chervonenkis (1971) amongst others.

Claim: Let x be any finite set and let H be a set of subsets of X. If k is the size of the largest subset of X shattered by H, then

$\backslash H \backslash \leq$ (IXI+l)*

Proof: By induction on Kl, the size of X.

Basis: Clearly true for $\backslash X \backslash = 1$.

Induction: Assume the claim holds for 1X1 = m and prove true for m+l. Let 1X1 = m+l and let H be any set of subsets of X. Also, let k be the size of the largest subset of X shattered by H. Pick any *xe x* and partition X into two sets {x} and Y» X-{x}. Define H_x to be the set of all sets in H that are reflected about x That is, for each set h_x in H_x, *there* exists a set A € H such that h dffers from h_x only in that h does not include x. Formally,

H_x = {$h_x \backslash$ h_x € H,3 h€ $H.h{*}h_x$ SMTd h_x - *u{x}}.

Now define H_2 = //-//^ Surely, the sets of $//_2$ can be distinguished on the elements of Y. That is, no two sets of H_2 can differ only on x, by virtue of our definition of H_v Hence, we can consider H_2 as sets defined on Y. Surely, H_2 cannot shatter a set larger than the largest set shattered by H. Hence, H_2 shatters a set no bigger than k. Since in £ /», by the inductive hypothesis we have $\backslash H_2 \backslash \leq$ (!7l+r/.

Now consider $//_r$ By definitton, the sets of H_x are all distinct on Y. That is, for any two distinct sets $*_{It}$ ^ in $i/_{If}$ AjnK ^ Z^^^- Suppose H_x shattered a set S c Y_f IS! ^ L Then, $//$ would shatter 5u{x}. But, '«5o{xj!> it+1, which is impossible by assumption. Hence, H_x shatters a set of at most (&-1) elements in r. By the inductive hypothesis, we have

s/fji $\leq (|Y|+1)^{k-1}$.

Combining the two bounds, we have

$|H|$ » $\backslash H\text{-}H_X \backslash + \backslash H_X \backslash - \backslash H_2 \backslash + \backslash H_X \backslash$

$\leq (|Y|+1)^k + (|Y|+1)^{k-1} \leq (m+1)^k + (m+1)^{k-1}$

$\leq (m+1)^{k-1}(m+2) \leq$ (m+2)* £(1X1+1)*.

Thus the claim is proved.*

Returning to the lemma, we see that if X is all strings of length n or less on the binary alphabet, txi = 2^{n+1}, By our claim, if the largest set shattered by F_m is of size k,

$|F_n| \leq (2^{n+1}+1)^k$.

Hence, $k \geq \log(|F_n|)/\log(2^{n+1}+1) \geq \dim(F_n)/(n+2)$.

Since k must be an integer, we take the ceiling of the right-hand side of the last inequality. This completes the proof of the lemma. •

We can now use this lemma to prove the main theorem of this section.

Theorem 1: A family F of concepts is feasibly learnable if and only if it is of polynomial dimension.

Proof: (If) Let F be of dimension d(n). The following is a learning algorithm for F, satisfying the requirements of our definition of learnability.

Learning_Argortthm A_X

Input: n, h
begin
call EXAMPLE H&mlFJlnil) + ln{h)) times.
let S be the set of examples seen.
pick any concept g in F consistent with S
output #.
end

We need to show that A_t does indeed satisfy our requirements. Note that A_x may not be computable, but as noted earlier, this is not a difficulty. Let/be the concept to be learned. Since P is a distribution on P[1], EXAMPLE returns examples of /„. We require that with high probability, A_x should output a concept g € F_t such that the probability that / and g differ is less than (I/A). Let $C_h(j)$ be all concepts in F_n that differ from/,, with probability greater than I/A. By definition, for any particular g such that g_K € $C_h(f)_t$ the probability that any call of EXAMPLE win produce an example consistent with g is bounded by (1-1/A). Hence, the probability that m calls of EXAMPLE will produce examples all consistent with g is bounded by (1-1/*)*. Ami hence, the probabBity that m calls of EXAMPLE will produce examples all consistent with any g_n € $C_h(f)$ is bounded by \C_k(f)Kl-l/hY". We wish to make m sufficiently large to bound this probability by 1 *fh.*

KT^1-1/Ar £ I/A.

Butsurely, \C_k(f)l ≤> IFJ£2^*>

Hence, we want

2**>(l-i/*r £ Vh

Taking natural logarithms on both sides of the inequality, we get

$d(n)\ln(2) + m\cdot\ln(1-1/h) \leq \ln(1/h)$

$-m\cdot\ln(1-1/h) \geq d(n)\ln(2) + \ln(h)$

$-m\,(-1/h) \geq d(n)\ln(2) + \ln(h)$

Or

$m \geq h(d(n)\ln(2)+\ln(h))$.

Heroe, f #dtn)fo(2Wn(k)) exarnpfes am drawn, the pmbabiiity *thsi* all the exam^es seen *am* consistent with a concept that differs from the true concept by I/A or more, is bounded by t/A. Since, A_t (taws as many examples and outputs a concept consistent with the examples seen, with probability l-l/A, A_x will output a concept that differs from the true concept with probability less than l/A. Hence, A_x does satisfy our requirements. Clearly, if *din)* is a polynomial in *n,* the number of examples called by A_1 is polynomial in n, *h* and hence *F* is feasibly learnable.

(only if)

Now suppose that *F* is of super-polynomial dimension *d(n)* and yet *F* were feasibly learnable by an algorithm *A* from $(nh)^k$ examples, for some fixed *k.* Let ¥ be the learning function corresponding to A. Now pick *n* and *h*≥ *5* such that

$$dim(F_n) \geq 2(n+1)(nh)^k.$$

By the shattering lemma, there exists a set *S* c I" such that ISI£ *dim(F^l(n+\),* and *S* is shattered by F_{rt}. Let X^le S^l denote the sequence x_v x^..., x_t and let/e F_n. Define the operator 8 as follows.

where $ »*¥fa h, (x^x^), *{x^x^^x^x^)),*

In words, S(*l*; X^7, ^F) is the probability error in the concept output by A on seeing the sample *{x/(x))*$_1$ (Jt2^x2))--(xi^))tor*l*- L6* G*n* ^F*n*be such that for each 5i ^5, there is exactly one *ge* G_n such that *gnS* = 5_r Such G_{rt} must exist as F_n shatters *S.* Let ^ be the probability distribution that is uniform on *S* and zero elsewhere.

Claim: Let *l* = *(nh)*.* Then for each *l*e G_w, and x*€ 5^, there exi^s unique g e G_{rt} such that $\delta(f,X^l,\Psi)$ ^ l/A if and only if 5(g.X^1,^) ^ 1/fc.

Proof: Let [X*} denote the set of strings occuring in X*, Le., {X*} = {^lx occurs in x^}. By the definition of *G_n,* for each *l*, X^l, there exists unique g e g_w » d i ttet/Ag * S-{X^}. Hence,

$$\delta(f,X^l,\Psi) + \delta(g,X^l,\Psi) \geq \sum_{x \in S-\{X^l\}} P(x)$$
$$\geq 1/2.$$

The last step follows from the fact that *{X?}* has at most half as many elements as 5, and *p* is uniform on S. Since A£5, *l/h* ^ 1/5, at most one of the terms on the left can be smaller than (1/5), if the inequality is to hold. Hence the claim. •

Since *¥ is a learning function for F, for each/€ F_n

$$\Pr\{\delta(f,X^l,\Psi) \leq 1/h)\} \geq (1-1/h)$$

(Notation: Pr{Y} derates the probability of event *Y.)*

Define the switch function 8:{true,false} -» M as follows. For any boolean-valued predicate fi,

MQ) m f 1* if G is true
t 0 otherwise

Now write

$$\Pr\{\delta(f,X^l,\Psi) \leq 1/h\} = \sum_{X^l \in S^l} \theta(\delta(f,X^l,\Psi) \geq 1/h)\Pr\{X^l\}$$

Substituting the above in the last inequality, we get,

$$\sum_{X^i \in S^i} \Theta(\delta(f,X^i,\Psi) \geq 1/h)Pr(X^i) \geq (1-1/h)$$

Summing over G_n,

$$\sum_{f \in G_n} \sum_{X^i \in S^i} \Theta(\delta(f,X^i,\Psi) \geq 1/h)Pr(X^i) \geq \sum_{f \in G_n} (1-1/h)$$

Flipping the order of the sums,

$$\sum_{X^i \in S^i} \sum_{f \in G_n} \Theta(\delta(f,X^i,\Psi) \geq 1/h)Pr(X^i) \geq \sum_{f \in G_n} (1-1/h)$$

By the last Claim,

$$\sum_{f \in G_n} \Theta(\delta(f,X^i,\Psi) \geq 1/h)Pr(X^i) \leq \sum_{f \in G_n} (1/2)Pr(X^i)$$

Hence, we have

$$\sum_{X^i \in S^i} \sum_{f \in G_n} (1/2)Pr(X^i) \geq \sum_{f \in G_n} (1-1/h)$$

Flipping the order of the sums again,

$$\sum_{f \in G_n} \sum_{X^i \in S^i} (1/2)Pr(X^i) \geq \sum_{f \in G_n} (1-1/h)$$

Which reduces to

$$\sum_{f \in G} (1/2) \geq \sum_{f \in G} (1-1/h)$$

which is impossible as $h \geq 5$.

The last contradiction implies that A cannot be a learning algorithm for F as supposed and hence the result.

This completes the proof.•

3. Learning Sets with One-Sided Error

We now consider a learning framework in which the learner is only allowed to see positive examples for the concept to be learned, and is required to be conservative in his approximation in that the concept output by the learner must be a subset of the concept to be learnt Historically, this was the framework first studied by [Valiant, 1984].

Let F be the family of concepts to be learned. EXAMPLE produces positive examples for some concept/ e F. Specifically, EXAMPLE produces a string *x* € */*. Let *P* be a probability distribution on £\ The probability that a string *x e l* is produced by any call of EXAMPLE is the conditional probability given by,

$$\frac{P(x)}{\sum_{x \in f} P(x)}$$

assuming the denominator is non-zero. If the denominator is zero, EXAMPLE never produces any examples. We can now define leamability as we (fid earlier.

Defn: A family of concepts *F* is *feasibly learnabfe with one-sided* amrI there exists an algorithm *A* such that

(a) *A* takes as inputs integers *n* and *h,* where *n* is the size parameter and *k* the error parameter.

(b) *A* makes polynomial^ few calls of EXAMPLE, polynomial in *n* and A. EXAMPLE returns positive examples for some concept / € F, chosen according to an arbitrary and unknown probability distribution *P* on 27[1]-. -

(c) For all concepts/ e Fand all probability distributions *P* on 27*~, with probability (1-1//0, *A* outputs *ge* F such that *gdf* and

$\sum$ F(x) <*k* I/A.

Defn: We say a family of concepts F is *well-ordered* if for ail n, F_n u 0 is dosed under intersection.

With these definitions in hand, we state and prove the following theorem.

Theorem 2: A family *F* of concepts is feasibly leamable with one-skied error, if and only if It is of polynomial dimension and is well-ordered.

Proof: (If) This direction of the proof begins with the following claim.

Claim: Let 5G2?*~ be any non-empty set such that there exists a concept *g* e F_m containing *s.* i.e. *gs* F_{mt} and *S<zg.* If F is weH-ordered, there exists a teasf concept/to F_n containing *g,* i.e.,

V #e F_M: $S \subseteq g$ implies $f \subseteq g$.

Proof: Let *S* c 2*~ be non-empty and let *If\J%-*)* be the set of concepts in F_m containing 5. Now the intersection of al these concepts /= {fir^n...}, is in F_m. To see this, notice that since $F_M u0$ *is* closed under Intersection,/€F_Mu0. But,/#0as£*0andSe/. Henoe,/€**„. •

This allows us to write the following learning algorithm for F.

Learnlng_A!gorfthm A_2

Input: n, h
begin
call EXAMPLE $h(d(nyin(2)+l^*(*))$ times.
let S be the set of examples seen.
output any g in F such that g_n is the least
concept in F_n containing S.
end

Let/be the concept to be learned. Since g_n is the least concept consistent with S, surely, g_n c /„- Using arguments identical to those used in our proof of Theorem 1, we can show that with probability greater than (l-l/*)$_f$ g will not differ from the concept to be learned with probability greater than i/h. This completes the "if" direction of our proof.

(only I) Let F be feasWy learnable with one-sided error by an algorithm A.' Let us show that F is well-ordered, i.e., for all n, $F_H u0$ is closed under intersection. Suppose for some *, $F_n u0$ were not closed under intersection, and that/, g were two concepts in $F_n u0$ such *thai fng* is not in F_n<j0. Now, surely *fng* * 0, and *hence fng* is not in F_n. Place the probability distribution that is uniform on *fng* and zero elsewhere on £"~, ami run the learning algorithm A for h » $2^{ft}+^{i}$. At each call of EXAMPLE, a randomly chosen element of *fng* win be returned. Since *fng* is not in F# A must fail to learn with one-sided error. To see this, suppose that A outputs some concept *eeF*. Now, since A claims to learn with one sided error, e_n*of*, if/were the concept to be learned. Similarly, $e_n Qg_t$ since g could well be the concept to be learned. Hence, e_n*ofng*. But since fel/2**^{1}, e_n must be *fng*, which contradicts the assumption *that fng* is not in F_n. By arguments similar to those of our proof of Theorem 1, we can show that F must be of polynomial dimension. An alternate proof is presented in [Natarajan, 1986]. Hence the claim. •

This completes the proof. •

We now exhibit a curious property of the well-ordered families. Specifically, we show that each concept (*except* the empty set) in a well-ordered family has a short and unique "signature".

For a weH ordered family $F_\%$ define the operator $M_n d?$*~^ F_n as follows.

$M_n(S) = \{$ tea^/e F_M such thai *Sof*, if such/exists

f n words, *kj$)* fe^1^^^^^^ in F_m consistent wihS.

Proposition 1: M_m is Wempotefit, I.e.,

$M_n(M_n(S)) = M_n(S)$

Proof: By the «teffr*kni of Af.$_t$ *MJ$)* :s the toast conc^t/e F_M such that *$of.* Sumly, *MJf)* =/ and hence the proposHon. *

Proposition 2: For $A \subseteq B \subseteq \Sigma^{*}$, I*MJA) BM* $M_m(B)$ *am* both defined, then

$M_n(A)$ C $MJB)$.

Proof: By the definition of M_n, B^$M_n(B)$. Since A c^, A^$MJB)$. Hence, $M_H(A)$ ^$M_n(B)_t$ by Proposition

Proposition 3: For A, B c l*. if $M_n(A)$ and Af^B) are defined,

$M_H(AKJB)$ = $M_n(M_n(A)KjM_n(B))$

Proof: Since A ci^CA), 5 $cM_H(P)_t$ AuB c $M_n(/LJuM_H(fi)$. Whence it follows from Proposition 2 that, $M_H(AuB)$ c $M_n(M_n(A)KjM_n(B))$. And then, since A c A u S, we have by Proposition 2

$M_n(A) \subseteq M_n(A \cup B)$

and similarly

$MJB)$ c $M_n(A \cup B)$

Hence,

$MJMuMJfl)$ c$MJ'A \cup B)$

Applying Proposition 2 again, we get

$M_n(M_n(A) \cup M_n(B)) \subseteq M_n(M_n(A \cup B))$

Applying Proposition 1 to the right-hand side,

$M_n\{M_n(AyjM_n(B))$ c $MJAKJB)$.

Hence, the proposition. •

With these supporting propositions in hand, we can show that every concept in F has a small "signature-.

Proposition 4: If F is weM-ordered, then for every /e F^/*0 there exists 5^2T", tsy $\leq dim(F_n)$, **such that** $f = M_n(S_f)$.

Proof: Let/ e F_n and let ^ be a set of minimum size such that/= $MJJSj)$. Consider any two distinct subsets S_v S_2 of S_f We claim that $\boldsymbol{M_n(S\{)}$ * $M_n(S£$. To prove this, we will assume the contrary and arrive at a contradiction. Suppose $M_n(S_x)$ » $M_n(\$£$ for S_x * S_2. Without loss of generality, assume I^I $\leq$!5$_2$L Now,

S_f* $(S_rS_2)uS_2$

Applying M_n to both sides,

$M_n(S_f) = M_n((S_f - S_2) \cup S_2)$

Applying Proposition 2 to the right-hand side, we get

$M_n(M_n(S_f - S_2) \cup M_n(S_2))$

Since $M_n(S_2) = M_n(S_1)$,

$M_n(S£$ $= M_n(M_n(S_f - S_2) \cup M_n(S_1))$

Applying Propositton 2 again,

$M_n(S_f) = f = M_n((S_f - S_2) \cup S_1)$

But K S ^ U S ^ < isry,

which contradicts our assumption that $S_f$was a set erf minimum size such that/a $M_m(S$^ H#nce$_f$ each distinct subset of S_f corresponds to a distinct / e F_m. (Notice that we have really shown that S_f Is shattered by F_n.) Which in turn implies that

$|F_n| \geq 2^{|S|}$

or

$dim(F_n) \geq |S|$

Hence the proposition. •

Conversely, we can show that Propositbn 4 is tight in the following sense.

Proposition 5: If F is well-ordered, there exists/e F_H such that $/= M_n(S)$ implies $|S| \geq dim(F_n)/(n+1)$.

Proof: A simple counting argument There are at most 2^{n+1} distinct examples. If every/€ F_n were definable as the least concept containing some set of d examples, then

$2^{(n+1)d} \geq |F_n|$ or

$(n+1)d \geq$ d M : ^ implying $d \geq dim(F_n)/(n+1)$.

Hence, the proposition. •

4. Time-Complexity Issues in Learning Sets

Thus far, we concerned ourselves with the information complexity of learning, i.e.,, the number of examples required to learn. Another issue to be considered is the time-complexity of learning, i.e., the time required to process the examples. In order to permit interesting measures of time-complexity, we must specify the manner in which the learning algorithm identifies its approximation to the unknown concept. In particular, we will require the learning algorithm to output a name of its approximation in some predetermined naming scheme. To this end, we define the notion of an index for a family of concepts.

In order for each concept in a family *F* to have a name of finite length, *F* would have to be at most countably infinite. Assuming that the family *F* is countably infinite, we define an *index* of *F* to be a function *itF* -» 2^r such that

V / ^ € *FJ * g* implies *I(f)rI(g)* = 0.

For each/€ *Fj(f)* is the set of indices for/.

We are primarily interested in families mat can be learnt efficiently, i.e., in time polynomial in the input parameters *, *h* and in the length of the shortest index for the concept to be learned. Analogous to our definition of learnability, we can now define polynomial-time leamability as follows. Essentially, a family is polynomial-time leamable, if it is feasibly learnable by a polynomial-time algorithm.

Defn; A family of concepts *F* is *polynomial-time learnable* in an index / if there exists a deterministic learning algorithm A such that

(a) A takes as input integers *n* and *h.*

(b) A runs in time polynomial in the error parameter *K* the length parameter *n* and in the length of the shortest index in / for the concept to be learned /. A makes poiynomially few calls of EXAMPLE, polynomial[4] in n, *h.* EXAMPLE returns examples for/ chosen randomly according to an arbitrary and unknown probability distribution *P* on I T .

(c) For all concepts / in *F* ami all probability distributions *P* on £"-, with probability (I-l//t) the algorithm outputs an index i_g *e I(g)* of a concept *g* in *F* such that

$$\sum_{x \in f \Delta g} P(x) \leq \text{V*}$$

We are interested in identifying the class of pairs (F, /), where *F* is a family of concepts and / is an index for it, such that *F* is polynomial-time leamable in /. To this end, we define the following.

Defn: For a family *F* and index /, an *ordering* is a program that

(a) takes as input a set of examples *S* = *{(x$_x$j$_t$)* , (x^j^-C0^-) such that**

x$_{It}$i2,X5~. € 2T, and 3^3^. € {0,1}.

(b) produces as output an index in / of a concept/ e *F* that is consistent with S_t if such exists, i.e., outputsiye /0forsome/€ Fsuchthat

$$\forall (x,y) \in S,\ y = f(x).$$

[4]*mmmtMf,* we could *pemitAto make* as 'many calls of EXAMPLE as *possltle* within its fme sound. Hits wl net change cur discussion substatiaiy, In *tm*'mtoBst of clahty w© wi net punua tife *s^mm^m.*

Furthermore, if the ordering runs in time polynomial in the length of its input and the length of the shortest such index, we say it is a polynomial-time ordering and *F* is *polynomial-time orderable* in *I*.

With these definitions in hand, we can state the following theorem.

Theorem 3: A family of concepts is polynomial-time learnable in an index *I* (1) if it is of polynomial dimension and is polynomial-time orderable in *I*. (2) only if *F* is of polynomial dimension and is random polynomial time orderable in *I*.[5]

Proof: (If) Let *Q* be a polynomial-time ordering for *F* in *I*. The following is a polynomial time learning algorithm for F in *I*.

Learning_Algorithm A_3

Input: n, h
begin
cal EXAMPLE *MdmFJ* + W) times;
let *s* be the set of examples seen;
output 2(5);
end

Given Theorem 1, we know that A_3 learns *F,* and only need bound its running time polynomial. Now, *Q* runs in time polynomial in the size of its input and the length of the shortest index of any concept consistent with *S.* Since the concept to be learned must be consistent with 5, surely *Q* runs in time polynomial in A, *h* and in the length of the shortest index of the the concept to be learned. Hence, A_3 runs in time polynomial in *n, h* and in the length of the shortest index for the concept to be learned. Therefore, *F* is polynomial-time leamabte in *I*.

(Only if) Assume that *F* is polynomial time Seamabie in an index *I* by an algorithm A. Since *A* calls for polynotniaily few examples, *F* must be of polynomial dimension by Theorem 1. It remains to show that there exists a randomized polynomial-time ordering for F. The following is such an ordering.

Ordering *0*
Input: *S:set* of examples, ^integer;

begin
place the uniform djstn'bution on *S;*
let *h* = |S|+1:
run A an inputs n, $*_f$ and
on each cal of EXAM FLE by A
return a randomly chosen element of 5.
output the index output by A.
and

Let *f* be a concept consistent with 5, whose index length is the shortest over all such concepts. Now, with probabffity (1-l/A) *A* must output the index of a concept *g* that agrees with/with probability greater

[5] A randomized algorithm at one that *mam cmm* cintng Ms cwnptftatiart and produces the correct answer with high probability.

than (I-I/A). Since the distribution is uniform and *h* > \S\, *g* must agree with/on every example in 5. Hence with high probability, *g* is consistent with *S.* Furthermore, since A is a polynomial-time learning algorithm for F, our ordering *o* is a randomized polynomial-time ordering for *F* in /. To see this, notice that *A* runs in time polynomial in *n* and A, and /, the length of the shortest index of/ By our choice of *, it follows that *A* runs in time polynomial in *, LSI and /. Hence, *O* runs in time polynomial in *, *h* and /, and is a randomized polynomial-time ordering for *F* in /.

This completes the proof. •

We can state analogous results on the time-complexity of learning with one-sided error. Specifically, an ordering for a well-ordered family would be an ordering as cfefined earifer with the exception that it would produce the least concept consistent with the input. Also, we can modify our definition of polynomial time leamability to allow only one-sided error. We can then state and prove the following.

Theorem 4: A family *F* is polynomial-time learnable with one-sided error; (1) if it is of polynomial dimension, well-ordered and possesses a polynomial time ordering; (2) only if it is of polynomial dimension, well-ordered and possesses a random polynomial time ordering.

Proof: A straightforward extension of earlier proofs. •

5. Learning Functions

in the foregoing, we were concerned with learning approximations to concepts or sets. In the more general setting, one may consider learning functions from 2T to r\ To do so, we must first modify our definitions suitably and generalize our formulation of the problem.

Defn: We define a family of functions to be any set of functions from 2T to I*. For any *fe* F_n, the projection/^:!*--*!* of/on Z^n is given by

$$f_n(x) = \begin{cases} f(x), \text{ if } |f(x)| \leq n \\ n\text{-length prefix of } x, \text{ otherwise} \end{cases}$$

Defn: The /^subfamily F_n of F is the projection of F on X^n, i.e,
$F_n = \{f_n | f \in F\}$.

The above two definitions are the analogues of the corresponding definitions for sets. The notion of the projection/„ of a function/attempts to capture the behaviour of/on strings of length *n.* If for some *xe* r^./Cx) is not of length at most *n,* it is truncated to *n* characters.

An example for a function/is a pair (xoO, *x^y eV* such that *y* =/fr). A learning algorithm (or more precisely a learning function) for a family of functions is an algorithm that attempts to infer approximations to functions in *F* from examples for it. The learning algorithm has at its disposal a subroutine EXAMPLE, which at each call produces a randomly chosen example for the function to be learned. The examples are chosen according to an arbitrary and unknown probability distribution *p* in that the probability that a particular example *{xj(x))* will be produced at any call is *P(x).*

As in the case of sets, we define leamability as follows.

Defn: A family erf functions *F* is *feastoty leamable* If there exists an algorithm *A* such that
(a) *A* takes as input integers *n* and *K* where *n* is the size parameter and *h* the error parameter.

{h)A makes polynomial few calls of EXAMPLE, polynomial in *n* and *h.* EXAMPLE returns examples for some function/^ *e* $F_{n\%}$ chosen according to an arbitrary and unknown probability distributions on Σ^n.

(c) For ail functions f_m e F_n and ail probability distributions *P* on *IT,* with probability (1-1//*), *A* outputs a a function *ge F such* that

$$\sum_{f_n(x) \neq g_n(x)} P(x) \leq 1/h$$

Our defratJoft of dimension *"m* this setting is exactly the same as the one given earlier for concepts. We ran now generalize the notion of shattering as follows.

Dtfti: Let *F be* a family of functions from a set *X* to a set r. We say *F shatters* a set *$cX* I there exist two fyncifons/f$ € Fsuchthat
(a) for any *sc SJ(s)*t(f).*

(b) for al 5, c *S,* there exist *e* e *F mxh* thai *e* agrees with/on S_x and with *g* on SS_X. s.e.f

Vj e Sj :*(»=:/(s)
$\forall s \in S-S_1: e(s) = g(s)$.

We can now generalize our shattering lemma for functions as follows.

Lemma 2 (Generalized Shattering Lemma): if F_n is of dimension 4 F_n shatters a set of size *ceiling(d/@n+3))).* Also, every set shattered by F_n is of size at most *d.*

Proof: The upper bound part of the lemma can be proved exactly as the corresponding part of Lemma 1. To see that this upper bound can be attained, we simply need to consider a family F_n of {0,1}-valued functions.

The lower bound part of the lemma is proved through the following claim.

Claim: Let *X* and 7 be two finite sets and let *H* be a set of functions from *X* to 7. If *k* is the size of the largest subset of *X* shattered by *H,* then
$|H| \leq (|X|)^k(|Y|)^{2k}$.

Proof: By induction on IXL

Basis; Clearly true for 1X1=1, for alt 171.

Induction: Assume true for 1Y1 = /, 171= *m* and prove true for 1X1 = /+l, 171 = m. Let *X* = (x^ J^ .. ., x_l} and $Y = \{y_1, y_2 \ldots, y_l\}$. Define the subsets$H_i$-otHasfollows.
$H_i = \{f \mid f \in H, f(x_1) = y_i\}$.

Also, define the sets of functions H_{Lj} *ami* H_Q as follows.
for $i \neq j$: $H_{ij} = \{f \mid f \in H_i, \exists g \in H_j$ such that $f = g$ on $X - \{x_1\}\}$.
$H_0 = H - \bigcup_{i \neq j} H_{ij}$.

Now,

HI-lHy + iU^/r^i ^ Hy+ $\displaystyle\sum_{i \neq j} |H_{ij}|$.

We seek bounds on the quantities on the right-hand side of the last inequality. By definition, the functions In H_o am ail distinct on the *m* elements of *X-ixJ.* Furthermore, the largest set shattered in H_Q must be of cardinality *no* greater than *k. Hence,* we have by the inductive hypothesis,
$|H_0| \leq l^k m^{2k}$.
Ami then, every #&. shatters a set of cardinally at most * - l , as othifwtee *H* would shatter a set of cardinality greater than *L Nm* *skim* the functions *in* H_9 are a! dstinct on x - {xj}$_f$ we have by the Inductive hypothesis,
For $i \neq j$, $|H_{ij}| \leq l^{k-1} m^{2(k-1)}$.

Combining the last three inequalities, we have

$$|H| \le l^k m^{2k} + \sum_{i \ne j} l^{k-1} m^{2(k-1)} \le l^k m^{2k} + m^2 l^{k-1} m^{2(k-1)} \le l^k m^{2k} + l^{k-1} m^{2k}$$

$$\le m^{k-1} l^{2k}(m+1) \le (m+1)^k l^{2k}.$$

Which completes tfie poof of the daim. •

Returning to the *mm*.m*tomX-Y-ir,a**ml-M-1!»'.* If *k* is the cardinality of the tagest set in I?" shattered by Λv we have by our daim,

$$\le (2^{n+1})^k (2^{n+1})^{2k}$$
$$\le 2^{k(3n+3)}.$$

Taking logarithms,

$$\le k(3n+3)$$

Hence, k £ 4/(3/t+3), which is as desired- •.

Using this lemma, we can prove the following theorem.

Theorem 5: A family of functions is feasfoly learnabte if and only if it is of polynomial dimension.

Proof: Similar to the proof of Theorem 1, except that we need use the generalized notion of shattering and the corresponding generalized shattering lemma. •

Analogous to our development erf time-complexity considerations for concept learning, we define the following.

For a family of functions *F* of countable cardinality, we define an index / to be a naming scheme for the fundtona in *FJn* a sense identical to that for a famBy of concepts.

We say a family of functions *F* is *polynomial-time leamable* in an index /, if there exists a deterrnmistic Earning aigorithm .4 such that

{a) *A* takes as input integers *n* ami *h*.

(b| *A* runs in time pc^nomisi in *the mm* parameter A, the length parameter *n* and in the length of ft* shortest Index in / for the function to be teamed / A makes polynomial^ few calls of £XAMFt£f *po^mmM* in «. *tu* EXAMPLE rrturm example *forf*$_n$ chosen ranctomly according to an arbitrary swd unhncwm prabability **distribution *P* on Σ*.**

(c) Rr a1l concepts / In F and aH pnbobHy (I^ta^H^ F on 27[1], with probability (1-I/A) the a _ n *F* **such that**

$$\sum_{f_n(x) \ne g_n(x)} P(x) \le 1/h$$

We am Wtftsted fciktotiyir^ tht dass of pafcs (f_f/), where *F* te a famity of concepts ami / is an **index for it,** start **that *F* is polynomial-time leamable in *I*.** To this end, we define the following.

Qitt: **For a family *F* and index *I*, an *ordering* is a program that**

fal trim as input a set of *msmplm S* » *{(x*$_t$*^*$_t$*)*$_f$ (^ %)— (WiM- Let « ^tt[1]©^^th of the kmgast rt*^ among tht x_i and |^

(b) produces as output an index in / of a concept/ € *F* that is consistent with *S,* if such exists, i.e., outputs *if* € /(/) tor some/ € *F* such that

$$\forall (x,y) \in S,\ y = f_{\kappa}(x).$$

Furthermore, if the ordering runs in time polynomial in the length of its input and the length of the shortest such index, we say it is a polynomial-time ordering and *F* is *polynomial-time orderable* in /.

With these definitions in hand, we can state the following theorem.

Theorem 6: A family of functions is polynomial-time leamable: (1) if it is of polynomial dimension and polynomial-time orderable; (2) only if it is of polynomial dimension and is orderable in random polynomial time.

Proof: Similar to that of Theorem 3. •

6. Finite Learnability

Thus far we explored the asymptotic learnability of families of sets and functions, that is to say, we considered the asymptotic variation of the number of examples needed for learning with increasing values of the size parameter. We will now investigate a different notion of learnability, one that asks whether the number of examples needed for learning is finite, Le, varies as a finite-valued function of the error parameter, without regaid to the size parameter. We call this notion of leamability "finite learnabilrty" as opposed to the notion of asymptotic learnability.

For the case of families of sets, [Blumer et at, 1986] present conditions necessary and sufficient for finite-leamability. Their elegant results rely on the powerful results in classical probability theory of [Vapnik and Chervonenkis, 1971]. In the following we review their results briefly and then go on to present leamability results for families of functions, relying in part on the same results of [Vapnik and Chervonenkis, 1971].

Defn: Let *F* be a family of sets on R*, where R is the set of reals and £ is a fixed natural number. We say *F* is *finitely learnabfe* if there exists an algorithm *A* such that

(a) *A* takes as input integer *h,* the error parameter.

(b) *A* makes finitely many calls of EXAMPLE, although the exact number of calls may depend on A. EXAMPLE returns examples for some function/in *F,* where the examples are chosen randomly according to an arbitrary and unknown probability distribution *P* on R.

(c) For all probability distributions *P* ami all functions/in F, with probability (1-1/A), *A* outputs *geF*

$$\int_{f*g} dP \; £ \; 1/k$$

The following theorem is from [Blumer et al., 1986].

Theorem 7: [Blumer et ai., 1986] A family of sets F on R* is finitely leamable if and only if *F* shatters only finite subsets of R*. ([Blumer et al., 1986] refer to the size of the largest set shattered by *F* as the *Vapnik-Chervonenkis dimension* of the family F).

Let us now formalize the notion of finite leamabiiity of families of functions on the reals.

Defn: Let F be a family of functions from R* to R*, where R is the set of reals and *k* is a fixed natural number. We say *F Is finitely leamable* if there exists an algorithm *A* such that

(a) *A* takes as *input* integer A, the error parameter.

(b) *A* makes flniely many calls of EXAMPLE, although the exact number of calls may depend on *h.* EXAMPLE returns examples for some function/in F, where the examples are chosen randomly according to an arbitraiy ami unknown probability distribution *P* on R*.

(c) For all probability distributions *P* and all functions/ in F, with probability (1-1/A), *A* outputs *g€ F* such that

$$t \; dP \; £ \; Vk$$

We need ttwfoicming support^ deffn«ion& Let/be a function from R* to R*. We defJne the *graph otf,* denoted *by gropMf),* to be the set of ai examples for/. That is, *graph(f)* m {(xj)l*y* =/(*)}.

Clearly, *grapMf)* c R*xR*. Analogously, for a family of functions F, we define *grapHF)* to be the set of graphs for the functions in F. That is,

graph(F) = {graph(f)\f € F}.

We now state the main theorem of this section. The theorem is not tight in the sense that the necessary and sufficient conditions do not match. (In [Natarajan, 19881, a tight version of the theorem was reported, on the basis of an incorrect proof.) Indeed, we will identify a finitely leamable family of functions that sits in the gap between these conditions.

Theorem 8: A family of functions F from R* to R* is finitely learnable

(a) If there exists a bound on the size of the sets in R*xR* shattered by *grapk(F).* (simple shattering as defined in Section 2.)

(b) Only if there exists a bound on the size of the sets in R* shattered by F. (Generalized shattering as defined in Section 5.)

Proof: (If) This direction of the proof follows from the convergence results of [Vapnik and Chervonenkis, 1971] exactly as shown in [Blumer et aL, 1986]. Essentially, the TP condition implies that the family *graphij)* is finitely leamable. Whence it follows that the family *F* is finitely leamable.

(Only if) This direction of the proof is identical to the asymptotic case of Theorem 4, which in turn followed the arguments of Theorem 1. •

While Theorem 8 is not tight, it appears that tightening it is a rather difficult task. Indeed we conjecture that the "if" condition should match the "only if condition as stated below.

Conjecture: A family of functions *F* from R* to R* is finitely leamable if ami only if there exists a bound on the size of the sets in R* shattered by *F.*

To give the reader a flavour of the difficulties involved in tightening Theorem 8, we give an example of a family F of functions that lies in the gap between the necessary ami sufficient conditions of Theorem 8, Le

(a) F shatters sets of size at most one.

(b) *grapWF)* shatters arbitrarily large sets.

(c) F is finitely leamable.

Example: Let M be the natural numbers in binary representation. For any ae N, define the function l_a:N-»N as follows.

$$f_\alpha(x) = \begin{cases} \text{a, if the*}^{1}\text{* bit of a is 1} \\ \text{0 otherwise} \end{cases}$$

Define the family *F* as follows.

$F = \{f_\alpha | \alpha \in N\}$.

Claim: F shatters sets of size at most one.

Proof: Suppose F shatters a set of size greater than one. Then F must shatter a set of size 2. Let *S * {ajb}* be such a set. By definition, there exist three functions/ g_9 *e* in F such *\haif(a)*g(a),f(b)*g(b)* and *e(a) **f(a\ e(b)« g(b).* Since, *fta)*g(a)$_t$* one of them must be zero and the other non-zero. Without toss of generality, assume thatyfo) is non-zero. Mow, by the definition of the functions in /\/(a) = *(a) * 0 implies that/= *e.* This contradicts the assumption that *e(b)« g(b) *f{b)>* and hence the claim. •

Claim: *graph{F)* shatters arbitrarily large sets.

Proof: Let S_l be any arbitrarily large but finite subset of N. Consider $S = S_x x\{0\}$. It is easy to see that *grapHF)* shatters S_f as for any subset S_2 of *S,* there exists a set/ e F such that/n 5 = s_2. To see this, notice that for any subset Sj of S, we can pick an integer *ae* N, such that/$_a$ n 5 = ^ Since *S* was picked to be arbitrarily large, the claim is proved. •

Claim: F is finitely iearnable.

Proof: The following is a learning algorithm for *F.*

Learning Algorithm A_4
Input *h;*

begin
call for «0#(ft) exanples.
If any of the examples seen is of the
form (x,y), y≠0
then output*f*,
else output/Q.
end

It is easy to stow that the probabilities work out for algorithm *A above.* Suppose the function to be learned were/$_a$, for some a#0. Then, if

$$\int_{f_\alpha \neq f_0} dP \leq l/A,$$

with probability (1—l/A), in *hlogh* examples there must *be* an example of the form (x,a). In which case, the algorithm will output/^, implying that with probability (1-i/A), the algorithm learns the unknown function exactly. Hence the claim. •

The interesting thing about the functions fet F is that each function (Mere from the 'base function/$_0$ on fHttfy iwiy jx*«s$_t$ and on ttiase poite, the vsAie of the function is the name of the function. Berm if th» teaT*^ a^rWwn sees a non-zero value h sm example, i *cm* urA^jely Wentify the function be teamed* •

Tins far. *we cmmkimmi fwuHkm* on *real* spaces, requiring that on a randomly chosen poirt$_v$ **with** Mgh 'prolMi»y tha iaamar's application sp«e exactly wih the function to be learned. TWs **requires** infinite precision arithmetic and hence is largely of technical interest. But then, if all the computations are carried out only to some finite precision, Theorem 5 would apply directly. Alternatively, we could require that the learned function approximate the target function with respect to some predetermined norm. In the following, we consider the case of the square norm, for a single probability distribution P.

First, we limit the discussion to families of "normalized- functions. Let *E(aJ>)* denote the euclkJean distance between any two points *a* and *b*. Let F:R*->R* be a family of functions such that for every *fe F* and jce R*, £(/(x),0*) ^ 1, where 0* is the origin *\n* R*. Then, we fix the probability distribution *P.*

Defn: We say that F is finitely learnable with respect to the square norm and a distribution *p am* R*, if there exists an algorithm A such that:

(a) A takes as input an integer /t, the error parameter.

(b)A makes finitely many calls of EXAMPLE, though the exact number may depend on *h.* EXAMPLE returns examples for some function/in F, where the examples are chosen according to the distribution *P.*

(c) For all functions/ € F, with probability /t, A ou^Hits a function *g e F* such that

$$\int_{x\in} E(f(x),g(x))dP \leq 1/h.$$

Before we can state our result in this setting, we need the following defintkm, adapted from [Benedeck and ttaf, 1988].

Defn: For small positive 5: *K&F* is a 5-cover with respect to the square norm and distribution *P* if, for any *fe F* there exists g€ *K* such that,

$$\int_{x\in R^k} E(f(x),g(x))dP \leq \delta$$

Theorem 9: A family of functions is finitely iearnable with resçec: to the square norm and a distrtoution P_t if and only »for all positive 8, there exists a ftniie 8-cowrfor F.

Proof: The details of the pioof are Identical to that of the main theorem of {Bemcfeek actd ill, 19881. A teaming algorithm' *A* for F can be described as *toSoms:* on Input A, *A* constructs an *iik<mm oi F* of minimum size. *A* then calls for sufficiently many examples to pem* I to fAdc on® erf the functions «n the knot with sufficiently high confidence. •

7. Acknowledgemerits

I thank R.Kannan, D.Sleator, P.Tadepail and T-MMchett tor nwny useful discussions.

$. References

[I] Angluin, D., and Smith, C.H., (1983). Inductive Inference: Theory and Methods. *Computing Surveys, 15(3.* (pp. 237-269).

[2J Angluin, D., (1986). Learning Regular Sets from Queries and Counter-Examples, Tech. Report, YALEU/DCS/TR-464.

[3] Benedeck, G.M., and Itai, N., (1988). Learning by Fixed Distributions. *Proceedings of the Workshop on Computational Learning Theory,* (pp. 80-90).

[4] Berman, P., and Roos, R., (1987). Learning One-Counter Languages in Polynomial Time. In *Proceedings of the Symposium on Foundations of Computer Science,* (pp. 61-67).

[5] Blumer, A.,f Ehrenfeucht, A, Haussler, D.,f & Warmuth, M., (1986). Classifying Learnable Geometric Concepts with the Vapnik-Chervonenkis Dimension., In *Proceedings of the ACM Symposium on Theory of Computing* (pp. 273-282).

[6] Kearns, M., U.M., Pitt, L, and Valiant, L.G., (1987). On the Leamability of Boolean Formulae. In Proceedings *of the ACM Symposium on Theory of Computing,* (pp. 285-295)..

[7] Laird, P., (1987) Learning from Data Good and Bad, Ph.D Thesis, Dept. of Computer Science, Yale University.

[8] Michalski, R., Mitchell, T., and Cartoned, J.? *Machine Learning: An Artificial Intelligence Approach,* Tioga Press, Palo Alto, CA.

[9] Natarajan, B.K., (1986). On Learning Boolean Functions. Carnegie-Mellon Robotics Institute TR-86-17 and in *Proceedings of the ACM Symposium on Theory of Computing, 1987.* (pp. 296-304).

[10] Natarajan, B.K.,f (1988). Two New Frameworks for Learning, *Proceedings of the Fifth International Symposium on Machine Learning,* (pp. 402-415).

[II] Rivest, R.,f and Schapire, R.E., (1987) Diversity Based Inference of Rnite Automata. In *Proceedings of me Symposium on Foundations of Computer Science,* (pp. 78-87).

[13] Valiant, LG., (1984). A Theory of the Leamabte. In *Proceedings of the ACM Symposiim on Hmwy of Computing,* (pp.436-445).

[14] Vapntk, V.N., and Chervonenkis, A.Ya., (1971). On the Uniform Convergence of Relative Frequencies, Theory *of* Probability and its Applications, vol16, No2, (pp.264-280).